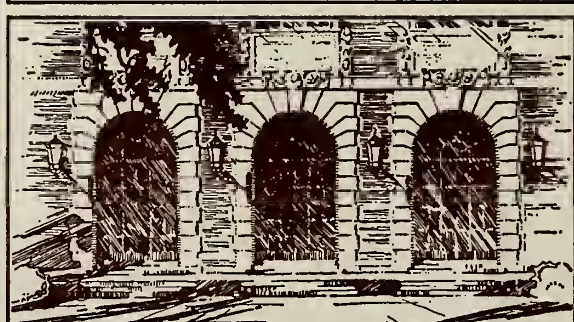

[redacted]

DCS Report No. 456

Math

ILLIAC IV

QUARTERLY PROGRESS REPORT

January, February, and March 1971

Contract No.
USAF 30(602)-4144

ILLIAC IV Document No. 247

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

ILLIAC IV

QUARTERLY PROGRESS REPORT

January, February, and March 1971

Contract No.
USAF 30(602)-4144

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois
61801

April 15, 1971

This work was supported in part by the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, and in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center, under Contract No. USAF 30(602)-4144.

TABLE OF CONTENTS

REPORT SUMMARY

Automation Technology, Inc. reports that PE tests and Card tests proceeded well during the quarter. Diagnostic efforts will be concentrated on "diagnosing" conversion problems through a major portion of the next quarter.

During this quarter the ILLIAC IV Project has received a Burroughs B6500/B6700 Computer at the University of Illinois. This machine has been obtained primarily to support the programmers developing the ILLIAC IV operating system. It is also available to applications programmers and to prospective users of the ILLIAC IV computer.

The software effort was directed toward integrating the Operating System with the ILLIAC IV simulator on the B6500, converting the Cockroach-to-Glypnir translator to the B6500, supporting the assembler (ASK) on the B6500, and simulating ILLIAC IV special functions and algorithms written in ASK for a mathematical subroutine library. The Software Reference Manual is nearly complete.

Several major software efforts for the PDP-11 system were continued or completed. Hardware delivery began on the various segments of the ARPA network terminal system. Negotiations were completed and agreement reached with Digital Equipment Corporation on a cooperative research arrangement for joint development of the ARPA network port facility.

Application efforts in Numerical Analysis continue in solving partial differential equations, matrix inversion and linear algebraic equations, eigenvalues and eigenvectors, estimation and filtering, and identification of nonlinear differential equations.

Three seminars were offered during the quarter.

Project expenditures and commitments through March, 1971:

Burroughs Corporation	$26,723,000.00
University of Illinois	6,814,273.83

1. HARDWARE

1.1 Off-Line Diagnostics

1.1.1 PE Test

The analysis of the PE logic and the development of tests to be used to test the PE off-line through use of the PE Exerciser (PEX) is progressing well. The Path Tests and Combinational Logic Tests are completed and in use. The analysis necessary for development of tests of the Control Logic is nearing completion and some test sequences have been defined. This area is expected to be complete by late summer.

1.1.2 Card Test

The development of tests to isolate stuck-type faults in active devices is on schedule except as noted below. All tests for PE Cards are complete, have been verified, and are in use. Of the CU Cards, the Type 1 boards are scheduled to be completed by mid-summer. The Type 1 boards have been divided into two classes, 31 Priority Boards and 40 Non-Priority Boards. Twenty-eight Priority Boards and 21 Non-Priority Boards have their tests generated; however, these tests are not verified, nor are they in use, since there is no CU Card Tester operational.

1.2 Program Conversion

Conversion of the programs used to develop the diagnostics mentioned above was begun during the quarter. Because of the importance of conversion to be able to carry on useful test generation, some other scheduled work is being postponed. The level of effort on the PE Control Logic Tests is temporarily reduced to analysis only and no board tests are being generated. The majority of diagnostic efforts will be concentrated on "diagnosing" conversion problems through a major portion of the next quarter.

1.3 ILLIAC IV Maintenance

ATI maintenance engineers are actively participating in the on-line debugging of ILLIAC IV. Among the areas of active participation have been: PE Board Test, PEM/MLU Test, PE Test, CU Test and I/O Test. In addition, ATI has identified the long lead spare parts and estimated the required number of each. Under authorization of the University of Illinois, ATI is obtaining competitive prices, where possible.

1.4 Financial

At this point in the contract, ATI is approximately ten percent (10%) below the budgeted cost estimates.

2. SOFTWARE

2.1 Operating System

2.1.1 Operating System I

The operating system is at present being integrated with the ILLIAC IV simulator (SSK) on the B6500, and has reached the stage where simple ILLIAC IV programs have been run under the control of the ILLIAC IV operating system. Instrumentation and error diagnostics are now being placed in some of the modules to allow the performance of the operating system to be measured in detail. An Operating System Maintenance Manual has been written.

2.2 Compilers and Translators

2.2.1 Glypnir

Glypnir, Version II, has been undergoing further consolidation. I/O for ILLIAC IV (as opposed to Simulator I/O) and Macro Facilities in the form of a pre-processor have been provided. The Glypnir compiler has also undergone a basic measurement investigation with a view to speeding it up. The possibility of providing a facility for separately compiling subroutines has been and is still being studied. A Glypnir Compiler Maintenance Manual has been written.

2.2.2 Cockroach

During the quarter the Cockroach-to-Glypnir translator was completed for a subset of the specified language on the B6500 and converted to the B6500. Subroutines and functions will be completed in the beginning of the second quarter. User documentation of the available features was also provided. The addition of an hourly employee to the staff hastened the completion and provided for more extensive debugging.

Cockroach is now available to users, and a certain limited amount of user support also is available.

2.3 Assembler

The Assembler (ASK) is now supported on the B6500. It compiles at about 1200 cards per minute. A plan for increasing its compiling speed to 2500 cards per minute has been formulated and is being implemented.

2.4 Interactive Communications and Graphics

2.4.1 Interactive Communications

During the reporting period several major software efforts for the PDP-11 system were continued or completed.

1. A high-level language compiler, PPESPOL, was completed. Version II is running on the B6500.
2. Version I of the ARPA Network Terminal System, ANTS, was completed. Basic portions of the system were checked out and actual runs made on the PDP-11.
3. Design of the link-up between Paoli and the University of Illinois, via the ARPA network, using two ANTS systems was completed and initiated.
4. Final specifications were agreed upon by Burroughs for the construction of an interface between the B6500/B6700 system and the ARPA network IMP.

During this reporting period, hardware delivery began on the various segments of the ARPA network terminal system. The basic PDP-11 processor, the 16K words of memory, the real-time block, the high-speed paper tape, ASR35 operator's teletype, four 2400 Baud line interfaces with adapters, one dataset control and interface for a Bell 103A dataset, plus two general-purpose interfaces to be used to connect the Gould Electrostatic Plotter to the system and to connect the Computek storage scope system were received. The remainder of the items ordered should be delivered within the next reporting period.

Two interfaces between the ARPA network IMP and the PDP-11 were received. Both interfaces were debugged, checked out, and installed in the University of Illinois PDP-11 and the Paoli PDP-11. Success in getting on the network now hinges on the completion of system software development.

Negotiations were completed and agreement reached with Digital Equipment Corporation on a cooperative research arrangement for joint development of the ARPA network port facility.

2.4.2 Graphics

No work was done in the graphics area during this reporting period because of total lack of personnel. An effort was begun, toward the end of the period, to acquire a full-time professional in graphics who is expected to join the Center during the next period.

2.5 Mathematical Subroutines Special Function Library

This quarter's work has continued on the ILLIAC IV Special Functions and Algorithm subroutine library. The following set of routines written in ASK have been successfully simulated by the current version of the simulator and the results are good for 15 significant digits:

64-bit mode:
- sine and cosine
- tangent and cotangent
- natural logarithm
- square root
- arctangent (1 argument)
- arctangent (2 arguments)
- hyperbolic sine and cosine
- hyperbolic tangent
- gamma range(0,1)
- gamma range(0,∞)

32-bit mode:
- sine and cosine
- tangent and cotangent

- exponential
- natural logarithm
- square root

Algorithmic library:

- addition/subtraction of matrices stored straight or skewed
- multiplication of matrices stored straight
- transpose of a matrix stored straight

At present work is being completed for:

64-bit mode: • gamma (X) $0 < X \leq 57$

$57 < X < \infty$ and

$X < 0$

• error function: erf (X) and

LN gamma (X).

For matrix operations the set is being expanded to rectangular matrices stored striaght and skewed. The same set will also be made available in Glypnir. These routines will continue to be refined. Maintenance, distribution and programming assistance in using the mathematical routines is available.

2.6 B5500/B6500 Operations

During this quarter the ILLIAC IV Project has taken delivery of a Burroughs B6500 computer at the University of Illinois. This machine has been rented primarily to support the programmers developing the ILLIAC IV operating system. It is also available to applications programmers from the Center for Advanced Computation and future users of ILLIAC IV. The machine was installed in the basement of the Coordinated Science Laboratory and became operational around March 1.

2.7 Software Documentation

The Software Reference Manual is now at 900 pages, largely completed. Its one deficiency is an Assembler users manual, which is at present about half completed.

3. APPLICATIONS

3.1 Numerical Analysis

3.1.1 Partial Differential Equations

3.1.1.1 Numerical Solution of Problems in Hydrodynamics

The results of numerical experiments with the Brailovskaya, Dufort-Frankel, Cheng-Allen, Crank-Nicholson and Lax-Wendroff finite difference schemes on the Burgess equation have been presented in a formal report which is now in the process of being printed [1].

A code is being written and debugged for the two dimensional subsonic and transonic flow around a circular cylinder, as the next step in exploiting fully the techniques of parallel processing in the numerical computations of three-dimensional fluid and gas flow problems. In a polar co-ordinate system the governing equations in non-dimensional form are:

Continuity $$\frac{D\rho}{Dt} + \rho\nabla \cdot \vec{V} = 0;$$

Momentum $$\rho \frac{D\vec{V}}{Dt} + \nabla p = \sqrt{\gamma}\, \frac{M_\infty}{Re} \left(4/3\, \nabla\nabla \cdot \vec{V} - \nabla \times \nabla \times \vec{V}\right);$$

Energy $$\rho \frac{DS}{Dt} = \sqrt{\gamma}\, (\gamma - 1)\, \frac{M_\infty}{Re}\, \Phi + \gamma\, \frac{\sqrt{\gamma}\, M_\infty}{Re \cdot Pr}\, \nabla^2 (p/\rho)$$

$\vec{V}$ = velocity, p = pressure, S = entropy, γ = ratio of specific heats, M_∞ = free stream Mach number, Φ = dissipations terms, Re = Reynolds number, Pr = Prandtl number.

The computational region comprising the flow field is divided into a suitable number of mesh points, which is a function of computer

storage capacity, and the dependent variables, namely the pressure p, radial and tangential components of the velocity, i.e. u, v, and temperature, T, are calculated at each mesh point. The equation of state for a perfect gas is assumed. Initially the gas is assumed to be flowing with a uniform velocity and the physical properties are assumed uniform at each mesh point.

To gain flexibility and save time on runs while the program is being debugged, the viscous terms have been suppressed in the calculation, so that the flow is treated as being inviscid. At time t = 0, the condition of zero radial velocity is imposed on the cylinder boundary, a suitable time step is selected, and the finite difference representation of the governing equations is used to calculate the values of the dependent variables at succeeding times. The finite difference scheme, chosen for these experiments is the two-step, second order accurate, Richtmyer variation of the Lax-Wendroff scheme [2], which has been previously tested with the Burgess equation [1].

3.1.1.2 Algorithm Development

During this period several numerical methods have been investigated in cooperation with AMOCO Oil Company to solve the elliptic type partial differential equations. The semi-iterative block Jacobi method and ADI method were adopted and both were implemented in Glypnir. The first method worked correctly and was made into a general type subroutine. An iterative step takes about 600 microseconds on ILLIAC IV in the case of 21 x 21 mesh points. The second method is still being implemented. The storage scheme for this method is more complicated than the first method.

3.1.2 Matrix Inversion and Solution of Linear Algebraic Equations

During this quarter, a program to solve linear equations and perform matrix inversion has been written in ASK, and is presently being debugged. It uses the Gaussian Elimination method and is designed to handle full matrices up to 630 x 630 for Gaussian elimination and up to 475 x 475 for inversion. These large matrices are not core

containable and are therefore broken down into blocks of 64 columns, each block being processed independently of the others as far as possible.

The algorithm works in two stages, first the matrix A is decomposed into the product of lower and upper triangular matrices, L U.

The second stage uses this decomposition to operate on a right hand side to solve $Ax = b$, on a unit matrix to form A^{-1} or on an arbitrary matrix B to form $A^{-1}B$. To reduce the round-off error, 'partial pivoting' is employed (see e.g. Wilkinsen, J. H., The Algebraic Eigenvalue Problem), also the inner products of vectors are accumulated using double precision arithmetic.

Work is just beginning on the Conjugate Direction method which will be implemented in Glypnir. This should prove an efficient algorithm for solving $Ax = b$ when the matrix A is not given explicitly but a subroutine which computes Ax given the vector x is available.

3.1.3 Eigenvalues

Jacobi's Method for finding eigenvalues and eigenvectors of real symmetric matrices (including complex Hermitian matrices: Let $A = B + iC$ be a complex Hermitian matrix where B is real symmetric ($B = B^t$) and C skew-symmetric ($C = -C^t$), then the real symmetric 2n x 2n matrix

$$A^1 = \left[\begin{array}{c:c} B & -C \\ \hdashline C & B \end{array}\right]$$

can be an input to the Jacobi algorithm) has been coded in ASK and satisfactorily tested on the B5500 ILLIAC IV simulator.

Eberlein's Method (Jacobi-like algorithm) for normalizing real non-symmetric matrices has also been satisfactorily tested.

The results of both algorithms agree with the literature up to 14 significant digits. An error of order $0(10^{-15})$ is due to conversion routines from internal machine representation of floating points to their external decimal representation.

To have a means of comparison, two well established algorithms by H. Rutishauser [3] and P. J. Eberlein [4] have been combined in one B6700 ALGOL program. It is in a running condition and exists under the file name (completed version) WINFRIED/BERNHARD/PROC. This file is a separately compiled procedure. The name of the procedure is JACEIG (N,A,TS,D,ROT,TMX).

In using this procedure the user has an option

1) He may write his own main-program for I/O and procedure call, but he then has to go through the B6700 BINDER for interrelation between his file and the above file or,

2) He can use the file BERNHARD/JACOBI/EIGEN which does the I/O and calculation for him.

For case (2) the following control cards and input-data are required:

```
δ EXECUTE BERNHARD/JACOBI/EIGEN
δ BCL EBOR
<integer 1>, <integer 2>,
matrix-elements in real.
δ END
```

where δ is an invalid character like multipunch 1,2,3

<integer 1> = order of matrix

<integer 2> = TMX with

- TMX = 0 no eigenvectors are calculated and printed
- TMX > 0 the right eigenvectors are calculated and printed
- TMX < 0 the left eigenvectors are calculated and printed.

The matrix for which the eigenproblem is to be solved can be either real symmetric or real non-symmetric. If real symmetric then only the eigenvalues are printed in decreasing order. If the eigenvectors are to be printed (TMX $\neq$ 0), then this will leave the same order as the eigenvalues, i.e., to λ_j corresponds TS_j in the order of their appearance.

If real non-symmetric the total eigenvalue matrix is printed. If the eigenvalues are complex then the a_{ii} occupy the diagonal while the imaginary part occupies the off diagonal portions.

If $a_{ii} \pm i\, a_{ij}$ is the eigenvalue, the corresponding eigenvector $t_i \pm i_{(t_j)}$ appears in column (row) i and column (row) j.

3.1.3.1 Eigenvalues and Eigenvectors of Symmetric Tridiagonal Matrices

3.1.3.1.1 QR Algorithm with Origin Shifts

The QR algorithm for finding eigenvalues of symmetric tridiagonal matrices has been written in XALGOL, debugged and tested for different-size matrices. A short description of the algorithm follows.

A is a symmetric tridiagonal matrix with diagonal elements C_i and subdiagonal elements b_i. Any symmetric tridiagonal matrix A_t, (t = 1, 2, ...), may be expressed as

$$A_t = Q_t R_t \tag{1}$$

where Q_t is orthogonal and R_t is upper triangular of the form

$$\begin{bmatrix} p_1 & q_1 & r_1 & & & 0 \\ & p_2 & q_2 & r_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & r_{n-2} \\ & & & & \ddots & q_{n-1} \\ & & & & p_{n-1} & \\ 0 & & & & & p_n \end{bmatrix}$$

$$\text{Let } A_{t+1} = Q_t^T A_t Q_t \qquad \text{then} \qquad A_{t+1} = R_t Q_t. \tag{2}$$

For large t, A_t approaches a diagonal matrix, Diag $[\lambda_i]$, where λ_i's are the eigenvalues of the matrix, A.

To accelerate convergence of this algorithm, we let

$$A_t' = A_t - K_t I$$

where K_t is the eigenvalue of the 2 x 2 matrix

$$\begin{bmatrix} c_{n-1}^{(t)} & b_{n-1}^{(t)} \\ b_{n-1}^{(t)} & c_n^{(t)} \end{bmatrix}$$

which is closer to $C_n^{(t)}$. Then we decompose $A_t' = Q_t R_t$ and find

$$A_{t+1} = R_t Q_t + K_t I.$$

By repeating these two steps, the eigenvalues of the matrix, A, are immediately available.

A document describing the algorithm with flow chart is being written.

3.1.3.1.2 Inverse Iteration to Find Eigenvectors

A program to calculate the eigenvectors of symmetric tri-diagonal matrices with given eigenvalues has been written in XALGOL, debugged and tested. The algorithm is as follows:

A is a symmetric tridiagonal matrix with c_i as its diagonal elements and b_i as its subdiagonal elements. Its eigenvalues are given rather accurately and arranged in descending order.

Let vector $\vec{V}$ be taken as the initial vector. We solve

$$(A - \lambda I)\ \vec{X} = \vec{V} \tag{1}$$

followed by

$$(A - \lambda I)\ \vec{Y} = \vec{X} \tag{2}$$

then $\vec{Y}$ is the eigenvector corresponding to eigenvalue λ.

To make the algorithm more effective, we let

$$(A - \lambda I) = LU \tag{3}$$

where L is unit lower triangular and U is upper triangular, set $\vec{V} = L\vec{e}$ where $\vec{e}^{\,t} = (1, 1, \ldots, 1)$, then

$$U\ \vec{X} = \vec{e}. \tag{4}$$

Hence $\vec{X}$ can be determined by back-substitution. The matrix U is determined by Gaussian elimination with partial pivoting, i.e. eliminate the variables in their natural order, but select the row having the maximal coefficient of X_i as the pivotal row in the i-th stage. Similarly, we can find the vector $\vec{Y}$.

A document describing the algorithm with flow chart and numerical examples is being written.

3.1.4 Polynomial Root Finder

Actual coding of a root finder has not been done in ASK. However ALGOL codings have been debugged on the B5500. Presently changes are being made in these programs to run on the B6500.

In general, the algorithm has been outlined to be: (1) evaluate polynomial at various intervals to determine whether a change of sign takes place, (2) shorten these intervals as much as possible, (3) proceed on new intervals with Newton-Rapheson iterations. Also an ALGOL program using Sturm sequences to evaluate intervals is presently being debugged on the B6500.

3.1.5 Identification of Non-linear Differential Equations

Much is known about the numerical solutions of differential equations, however, almost nothing is known about the reverse problem, i.e., given some observed function of time, $x(t)$, can we find a differential equation of which $x(t)$ is a solution?

Such problems exist in economics, chemistry, medicine and many other fields, the equations involved often being nonlinear. In the past solution has been restricted to linear problems or problems involving the estimation of only a few parameters. This has been caused not so much by lack of solution techniques, but by inadequate computing power. ILLIAC IV can alleviate this problem through its high computing speed and the fact that the problem can utilize the array processor efficiently.

Since many different differential equations may have the same solution, it is necessary to restrict the problem to invariant equations of the form:

$$F(x,\ x',\ x'',\ \ldots,\ x^{(m)},\ f_1,\ \ldots,\ f_n) = 0 \tag{1}$$

where F is assumed to be known,

$$x^{(j)} \equiv \frac{d^t}{dt^j}\ x(t) \text{ and}$$

$$f_i \equiv f_i(x^{(j_i)}) \text{ are unknown functions}$$

or x or one of its derivatives. The problem is to develop an algorithm to determine the f_i .

E.G. $\quad x'' + f(x) = 0$

If x were given as $x(t) = \sin t$, we could derive $f(x) = x$. Let $\hat{x}(t)$ be the observed function, defined on the interval $0 \leq t \leq T$.

The solution to (1) depends on the functions f_i and the initial conditions of x and its first n - 1 derivatives at t = 0.

I.e., $x = x\,(x_0, x_0', x_0'', \ldots, x_0^{(m-1)}, f_1, \ldots, f_n)$

or more briefly $x = x(\underset{\sim}{x}_0, \underset{\sim}{f})$.

We define the error functional

$$E\,(\underset{\sim}{x}_0, \underset{\sim}{f}) = ||\, \hat{x} - x\,(\underset{\sim}{x}_0, \underset{\sim}{f})\,||^2$$

where

$$||\, y\, ||^2 = \int_0^T w(t)\, [y(t)]^2\, dt$$

for some weighting function, w. E is a measure of the difference between observed function and the computed one for some $\underset{\sim}{x}_0$ and $\underset{\sim}{f}$.

The problem thus reduces to finding $\underset{\sim}{x}_0$ and $\underset{\sim}{f}$ such that $E\,(\underset{\sim}{x}_0, \underset{\sim}{f})$ is minimized.

For the numerical solution, $\hat{x}(t)$ is assumed to be observed at discrete, equal intervals and the integrals replaced by summations. It is also assumed that these observations are contaminated by random "noise" so that minimizing E constitutes a least squares solution to the problem.

The functions f_i are approximated over the interval on which they are to be identified, [a,b], by a linear combination of fixed orthonormal functions

$$f(x) \approx \sum_{j=1}^{n} q_j\, \Phi_j\, (x)$$

where $< \Phi_i, \Phi_j >_1 = \delta_{ij}$, the Kroneuker delta function, the inner product $< g, h >_1$ being given by $< g, h >_1 = \int_a^b g\,(x)\, h\,(x)\, dx$. In this implementation the functions, Φ_j, will be splines.

The Solution Algorithm

For simplicity, we consider the problem

$$\frac{dx}{dt} = f(x)$$

Let $f(x) = \sum_{i=1}^{n} q_i \Phi_i (x)$ and $x(0) = x_0 = q_0$. Define the solution of

$$x' = f(x)$$
$$x(0) = x_0$$

to be $x_f(t)$. The dependence on x_0 being implicit.

Then $E(f) = E(q) = ||\hat{x}\ x||^2$

$$= \int_0^T [\hat{x}(t) - x_f(t)]^2\, dt.$$

When $E(q)$ is minimized, $\nabla q\ E = 0$

where $(\nabla q\ E)_i = \frac{\partial}{\partial q_i} E. \quad i = 0, 1, \ldots, n.$

A vector q such that $\nabla q\ E = 0$ may be found iteratively using the Conjugate Gradient Method [1] :

$$q_{n+1} = q_n + c_n\, p_n$$

where: $p_0 = - (\nabla q\ E)^{(0)}$,

$$p_{n+1} = - (\nabla q\ E)^{(n+1)} + b_n\, p_n\ , \tag{2}$$

$$(\nabla q\ E)^{(n+1)} = \nabla q\ E(q_{n+1}),$$

$$b_n = - \frac{< (\nabla q\ E)^{(n+1)}, (\nabla^2 q\ E)^{(n+1)} p_n >_2}{< p_n\ , (\nabla^2 q\ E)^{(n+1)} p_n >_2}\ ,$$

$$(\nabla^2 q\ E)^{(n+1)} = \nabla^2 q\ E\ (q_{n+1})$$

and $\nabla^2 q\ E$ is the matrix of second derivatives:

$$(\nabla^2 q\ E)_{ij} = \frac{\partial^2}{\partial q_i\, \partial q_j}\ E\ .$$

The superscripts indicate the iteration count. The inner product $< a,b >_2$ being defined by

$$< a,b >_2 = \sum_{i=0}^{n} a_i b_i .$$

At each iteration, c_n is chosen so as to minimize $E(g_n + cp_n)$ over all c.

The calculation of $\frac{\partial}{\partial q_i} E$.

$$\frac{\partial}{\partial q_i} E = \frac{\partial}{\partial q_i} \int_0^T w(t) [\hat{x}(t) - x_f(t)]^2 dt$$

$$= -2 \int_0^T w(t) [\hat{x}(t) - x_f(t)] . \frac{\partial x_f}{\partial q_i}(t) dt \qquad (3)$$

if we abbreviate $\frac{\partial x_f}{\partial q_i}$ by δx_i, then δx_i satisfies the so-called "sensitivity equations"

$$\frac{d}{dt} \delta x_0 = \frac{\partial f}{\partial x}(x_f) . \delta x_0, \delta x_0 (0) = 1$$

and $$\frac{d}{dt} \delta x_i = \frac{\partial f}{\partial x}(x_f) . \delta x_i + \Phi_i (x_f), \delta x_i = 0, i = 1, 2, \ldots, n .$$

representing the sensitivity of the equation $x' = f(x)$ to changes in x_0 and f respectively.

If we define $\Phi_0 (x) \equiv 0$, these may be combined to:

$$\frac{d}{dt} \delta x_i = \frac{\partial f}{\partial x}(x_f) . \delta x_i + \Phi_i (x_f) \quad i = 0, 1, \ldots, n . \qquad (4)$$

To determine $\nabla^2 q E$ we note that,

$$\frac{\partial^2 E}{\partial q_i \partial q_j} = \frac{\partial^2}{\partial q_i \partial q_j} \int_0^T w(t) [\hat{n}(t) - n_f (t)]^2 dt$$

$$= - 2 \int_0^T w(t) \frac{\partial}{\partial q_i} x_f(t) \frac{\partial}{\partial q_j} x_f(t) dt$$

$$- 2 \int_0^T w(t) [\hat{n} (t) - x_f(t)] x \frac{\partial^2}{\partial q_i \partial q_j} (x_f) (t) dt$$

$$= 2 \int_0^T w(t)\ \delta n_i\ \delta n_j\ dt$$

$$- 2 \int_0^T w(t)\ [\hat{n}(t) - x_f(t)]\ \delta n_{ij}^2\ dt \tag{5}$$

where we define $\delta n_{ij}^2 \equiv \frac{\partial^2}{\partial q_i\ \partial q_j} (x_f)\ (t)$.

By differentiating (t) with respect to q_j, $\delta^2 n_{ij}$ may be shown to satisfy the equation:

$$\frac{d}{dt} \delta^2 n_{ij} = \frac{\partial^2 f}{\partial n^2}\ \delta n_i\ \delta n_j$$

$$+ \frac{\partial f}{\partial n}\ \delta^2 n_{ij}$$

$$+ \frac{\partial}{\partial n}\ \Phi_i\ \delta n_j + \frac{\partial}{\partial n}\ \Phi_j\ \delta_{ni} \tag{6}$$

Since we require only ∇^2 q E explicitly, we need calculate only the components $\Psi_i \equiv \sum_{j=0}^{n} \delta^2 n_{ij}\ p_j$.

Multiplying (6) by p_j and summing over j gives:

$$\frac{d}{dt}\ \Psi_i = \frac{\partial^2 f}{\partial n^2}\ \delta n_i\ (\textstyle\sum\ \delta n_j\ p_j)$$

$$+ \frac{\partial f}{\partial n}\ \Psi_i$$

$$+ \frac{\partial}{\partial n}\ \Phi_i\ \ (\textstyle\sum\ \delta n_j\ p_j)$$

$$+ (\frac{\partial}{\partial n}\ \textstyle\sum\ \Phi_j\ p_j)\ \delta n_i \tag{7}$$

where all summations are from j = 0 to n. This reduces the number of differential equations to be solved from $(n + 1)^2$ to n + 1 .

Using (5) we obtain

$$[\nabla^2 q\, E\, p]_i = 2 \int_0^T w(t) \quad \delta\, n_i \; x \; (\sum \; \delta\, n_j \; p_j) \; dt$$

$$- 2 \int_0^T w(t) \; [\hat{n} \; (t) \; - \; x_f(t)] \; \Psi_i(t) \; dt \; .$$

Preliminary experiments indicate that this algorithm is rapidly convergent, only 5-10 iterations being required when a good estimate of f is available.

Implementation on ILLIAC IV

A specialized compiler is needed to perform the following tasks:

1. Accept the form of the equation to be identified.
2. By using symbolic differential techniques, form the sensitivity equations.

3.1.6 Estimation and Filtering

3.1.6.1 Non-linear Least Squares Problem

Work is proceeding on the parameter estimation of non-linear least squares problems. The numerical algorithms to be investigated are:

1. modified Gauss method
2. gradient method
3. variable metric method
4. factorization method

The implementation of algorithm (1) has been successfully completed and the ASK code for (2) is proceeding for which more accurate results are expected. Work has also begun on the variable metric method.

3.1.6.2 Numerical Solution of the Non-linear Matrix Riccati Equation

An eigenvector solution of the matrix Riccati equation relating to optimal control theory has been studied during this quarter. For the

cases in which the matrix of coefficients resulting in linear dynamic systems can be transformed to a diagonal matrix, there is a straight forward method for constructing a similarity transformation whether or not the eigenvalues are distinct.

3.2 Linear Programming

This quarter has seen the switch to the B6500 computer installation. Regrettably, the transition involved a degradation of the B5500 assembler-simulator support for a prolonged period, without comparable facilities being available elsewhere. This seven week stretch was used to bring LP-system documentation into usable shape.

The simulation that could be performed was directed wholly toward the INVERT package. There are indications that simulation will not permit sufficient code "execution" to debug the inversion routines, but work is continuing.

B5500 SETUP routines have been run on the new computer, but require modifications and extensive updating to reflect the increasing load of SETUP that the ILLIAC IV will perform. Experience with our B6500 indicates that only the most minimal processing should remain on this easily overtaxed facility. The future will see consideration being given to transferring more functions, including those--such as sorting--which are ungainly and difficult to code for the ILLIAC.

The problem of removing the solution and associated information is now being resolved. Initially these SETDOWN procedures will be executed as a separate program as it is estimated that abnormal terminations will be frequent. Again, as with SETUP, the majority of the work--such as rescaling and transforming data to B6500 representation--will be executed on ILLIAC IV.

Documentation was reviewed, rewritten, and reorganized with the intent of improving the clarity of the system's interconnections and interpresumptions. Not surprisingly, discrepancies were found. Further, refinement will be necessary, but will be deferred until after the code has been run on the prototype ILLIAC.

3.3 Long Codes

An analysis of the noise vectors used for previous studies disclosed an alarming degree of correlation between the supposedly independent random vectors. Consequently, an improved pseudo-random number generator of proven period and potency was substituted for the previously used ad hoc generator. All identification algorithms tested were more effective when used with the less correlated noise vectors.

A Kalman filter algorithm is being implemented to allow testing of the parameters obtained from the identification algorithms without knowledge of the true values of the parameters. If the innovation sequence of a Kalman filter using the estimated parameters as a model and using the observed values as input is a white noise sequence, identification can be considered successful.

The most recently developed correlation approach for identification showed considerable promise before its use was suspended for recoding to run on the B6500. The equations for this scheme, like most other relations that occur in identification processes, include both the covariance matrix of the plant noise and the covariance matrix of the observation noise, which are usually unknown. Manipulation of the correlation algorithm expressions indicates that it will be possible to obtain not only estimates of the parameters of the system, but also estimates of the two covariance matrices. It may be necessary to require the correlation matrix of the observation noise to be in the form $\sigma^2 I$ (I = Identity matrix), but this restriction can probably be relaxed.

3.4 Signal Processing

A Glypnir subroutine has been written which computes the fast Fourier transform of any real vector of length N, where N is a power of two and $N > 64$.

A previously written Glypnir autocorrelation program has been put into the form of a subroutine.

Programs are being written for use on the B6500 computer which will be used to thoroughly test previously written ALGOL procedures and Glypnir subroutines.

Work is continuing with Amoco Production Company on implementing some signal processing and partial differential equations algorithms on the ILLIAC IV.

3.5 Education

3.5.1 ILLIAC IV Seminars

In addition to the graduate course on ILLIAC IV, three seminars were offered this past quarter: A one-day seminar was given on January 5, 1971. The outline follows:

Seminar on ILLIAC IV

Background	9:00 - 9:30 am
Buffer, Pipeline and Multiprocessor	
Hardware Structure	9:30 - 12:00 n
A General Description of Array A Sample Problem A More Detailed Description of Array A General Description of I/O System	
Test/Repair Equipment and Diagnostics	1:30 - 2:00 pm
Physical Characteristics	
Software	2:00 - 4:30 pm
Programming Languages -- ASK, GLYPNIR, FORTRAN Operating System Utilities	
Applications	4:30 - 5:00 pm

On January 11-15, a one-week seminar was given to Pan American Petroleum. On March 15-19, another one-week seminar was given to employees at Ames Research Center, Moffett Field, California, where ILLIAC IV is to reside. Additionally a one and one-half hour overview on ILLIAC IV was presented to about 300 Ames employees. The outline for the one-week seminar follows.

A Seminar on the ILLIAC IV System

I. Background -- Conventional and Unconventional Organizations

- A. Why ILLIAC IV?
  - 1. Conventional Organization
- B. How to speed up the Operation-Design Philosophies
  - 1. Overlap
    - a. Buffer
    - b. Pipeline
  - 2. Replication
    - a. General Multiprocessor -- Distribute, Memory, ALU, CU
      - i. Recentralize Memory
      - ii. Recentralize ALU
      - iii. Recentralize CU -- basis for ILLIAC IV
  - 3. Both
- C. ILLIAC IV is a Vector Processor

II. Hardware Structure

- A. Organization Chart
- B. ILLIAC IV Array -- General Description
  - 1. Control Unit (CU)
  - 2. Processing Element (PE)
  - 3. Data Paths
    - a. Control Unit Bus (CU Bus)
    - b. Common Data Bus (CDB)
    - c. Routing Network
    - d. Mode Bit Line
- C. An Illustrative Problem
  - 1. DO 10 I = 1, N
    10 A(I) = B(I) + C(I)
    - a. N = 64
    - b. N < 64
    - c. N > 64
  - 2. DO 10 I = 2, 64
    10 A(I) = B(I) + C(I-1)
    - a. Skew at Compile Time
    - b. Skew at Execution Time
  - 3. DO 10 I = 2, 64
    A(I) = B(I) + A(I-1)
- D. ILLIAC IV Array -- A More Refined Description
  - 1. PE
    - a. RGD
  - 2. PU
  - 3. CU
    - a. ADVAST
    - b. FINST
    - c. MSU
    - d. TMU
    - e. ILA
- E. Another Illustrative Problem
  - 1. Laplace's Partial Differential Equation describing the Steady-State heat distribution on a slab
- F. Data Allocation

G. ILLIAC IV I/O System
   1. I/O Subsystem
      a. CDC
      b. BIOM
      c. IOS
   2. Disk File System
   3. B6500
      a. CPU
      b. Memory
      c. Multiplexor
      d. Peripherals
         i. Remote Terminals
      e. Laser Memory
      f. ARPA Network

III. Configurations at CAC and Paoli

IV. Diagnostics and Test/Repair Equipment

   A. IDIAP
   B. PEX
   C. PEMX
   D. CUCT
   E. Some Physical Characteristics
      1. Slides

V. Programming Languages

   A. ASK
      1. Background, Review, Notation, Conventions
      2. Sample Problems
         a. Summing an Array of Numbers
         b. Finding the Maximum Value in an Array of Numbers
         c. Matrix Multiplication
            i. Skewed Storage
         d. Temperature Distribution on a Slab
            i. Case 1: one temperature value per PEM
            ii. Case 2: eight temperature values per PEM
   B. GLYPNIR
   C. FORTRAN

VI. Operating System

   A. ICL
   B. Utilities

VII. Some Applications

4. ADMINISTRATION

4.1 Administration and Services

4.1.1 Financial Report

Actual expenditures and obligations for January-March 1971:

	January	February	March
Burroughs Corporation	$312,000.00	($109,000.00)	N.A.*
University of Illinois	183,220.34	240,118.42	$168,093.51

Expenditures and obligations to date through March 1971:

Burroughs	$26,723,000.00
University	6,814,273.83

Budgeted expenditures - 3rd quarter, fiscal 1971.

	January	February	March
Burroughs	$567,900.00	$298,643.00	$252,758.00
University	215,302.00	215,302.00	215,302.00

*Monthly status report not received from Burroughs as of 4/30/71, therefore figures for March are not available.

REFERENCES

[1] Rajan, S. "Numerical Solution of the Partial Differential Equations of Gas Dynamics." Center for Advanced Computation, University of Illinois at Urbana-Champaign, (in preparation).

[2] Richtmyer, R. D., and Morton, K. W. "Difference Methods for Initial Value Problems." Interscience Publishers, 1967. Pp. 354-365.

[3] Rutishauser, H. "The Jacobi Method for Real Symmetric Matrices." Handbook Series Linear Algebra, Numerische Mathematik 9, 1-10, (1966).

[4] Eberlein, P. J., and Boothroyd, J. "Solution to the Eigenproblem by a Norm Reducing Jacobi Type Method." Handbook Series Linear Algebra, Numerische Mathematik 11, 1-12, (1968).

THESES

Moreno, V. "A Logic Test Generation System using a Parallel Simulator." Master's thesis. Urbana, Illinois: Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

Parker, J. L. "Logic per Track Information Retrieval System." Ph.D. thesis. Department of Computer Science, University of Illinois at Urbana-Champaign, 1971.

DOCUMENTS

Hashimoto, A., and Stevens, J. "Path Cover of Acyclic Graphs." ILLIAC IV Document No. 239. ILLIAC IV Project, University of Illinois at Urbana-Champaign, (December 24, 1970).

Schuster, Stewart. "An Information Management and Analysis System for ILLIAC IV." CAC Document No. 1. Center for Advanced Computation, University of Illinois at Urbana-Champaign, (December 11, 1970).

Schuster, Stewart. "A Statistical System for ILLIAC IV." CAC Document No. 2. Center for Advanced Computation, University of Illinois at Urbana-Champaign, (December 11, 1970).

Stevens, James, Jr. "Matrix Multiplication Algorithms for ILLIAC IV." ILLIAC IV Document No. 231, Department of Computer Science File No. 855. ILLIAC IV Project, University of Illinois at Urbana-Champaign, (August 26, 1970).

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Department of Computer Science University of Illinois at Urbana-Champaign Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

ILLIAC IV QUQRTERLY PROGRESS REPORT
January, February and March 1971

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

January-March 1971 - Progress Report of the ILLIAC IV Project

5. AUTHOR(S) *(First name, middle initial, last name)*

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 15, 1971	34	4

8a. CONTRACT OR GRANT NO USAF 30(602)-4144 b. PROJECT NO. ARPA Order 788 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ILLIAC IV Document No. 247 DCS Report No. 456 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* RADC TR

10. DISTRIBUTION STATEMENT

Qualified requestes may obtain copies of this report from DCS.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Rome Air Development Center Griffiss Air Force Base Rome, New York 13440

13. ABSTRACT

See the Report Summary on Page 1 within the Report itself.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Hardware Diagnostics						
Compilers and Generators						
Glypnir						
Cockroach						
Graphics						
Numerical Analysis						
Linear Programming						
Operating System I						
Assembler						
Education						
Signal Processing						
Matrix Algebra						
Ordinary and Partial Differential Equations						
Administration of Computing Centers						